Army Research Laboratory

An Evaluation of the Relationship Discovery Service as a Company Intelligence Support Team Technology

by Sue Kase (Ph.D), Heather Roy, Elizabeth Bowman (Ph.D), and Mark Mittrick

ARL-TR-6065 July 2012

NOTICES

Disclaimers

Army Research Laboratory
Aberdeen Proving Ground, MD 21005-5066

ARL-TR-6065 **July 2012**

An Evaluation of the Relationship Discovery Service as a Company Intelligence Support Team Technology

Sue Kase (Ph.D), Heather Roy, Elizabeth Bowman (Ph.D), and Mark Mittrick
Computational Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
July 2012	Final	December 2011

4. TITLE AND SUBTITLE
An Evaluation of the Relationship Discovery Service as a Company Intelligence Support Team Technology

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Sue Kase (Ph.D), Heather Roy, Elizabeth Bowman (Ph.D), and Mark Mittrick

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
ATTN: RDRL-CII-C
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-6065

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The U.S. Army Research Laboratory (ARL) Computational Information Sciences Directorate (CISD) Tactical Information Fusion Branch (TIFB) conducted an investigation of the current state of development of the Relationship Discovery Service (RDS) platform as a potential technology for Company Intelligence Support Teams (COISTs). RDS is an integrated set of Web-enabled, service-oriented applications designed to assist in the analysis of soft intelligence data. The visual representation of social relationships and quantitative network centrality measures generated by RDS are intended to reduce the time and effort required for a novice intelligence analyst to organize and extract key information surrounding terrorist organizations and their activities. In military intelligence, the recent focus is on decentralizing analysis to the company level by implementing COIST primarily consisting of novice intelligence analysts. RDS was tested and evaluated internally by persons with little to no experience in intelligence analysis. The Ali Baba data set, originally created by the National Security Agency (NSA), was cleaned, modified, and validated for the purpose of testing. Testing and evaluation results of RDS are discussed in terms of system reliability and responsiveness, and improvements to functionality.

15. SUBJECT TERMS
Social network analysis, Company Intelligence Support Teams

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Sue Kase
Unclassified	Unclassified	Unclassified	UU	30	19b. TELEPHONE NUMBER (*Include area code*) (410) 278-9762

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

Acknowledgments

Tactical Information Fusion Branch Chief Mr. Mark A. Thomas provided valuable advice on the design of the end user testing for the Relationship Discovery Service (RDS) system, as well as contributing a substantial amount of time playing the role of a novice intelligence analyst.

INTENTIONALLY LEFT BLANK.

1. Introduction and Background

In the field of social network analysis, the Tactical Information Fusion Branch's Social Network Analysis Team focuses on the problem of indentifying patterns in soft data generated by many different sources. The growth of information available from online sites and the availability of multi-source sensors produce a corpus of data that, if sufficient data mining capabilities existed, could aid intelligence analysts in understanding terrorist intentions in near real time. One of our approaches to developing methodologies for analysis includes the Relationship Discovery Service (RDS) platform.

The RDS platform was developed under the Advanced All Source Fusion (A2SF) Technical Program Annex (TPA) between the Computational Information Sciences Directorate (CISD) and the Communication-Electronics Research, Development, and Engineering Center's (CERDEC) Intelligence and Information Warfare Directorate (I2WD). RDS is an integrated set of Web-enabled, service-oriented applications designed to assist novice intelligence analysts in the analysis of large amounts of soft intelligence data.

RDS is intended to address issues related to the problems of current asymmetric warfare. To rapidly understand the current situation, commanders require the ability to track high-valued individuals (HVIs) within their area of operations (AO). This situational understanding increases the commander's ability to anticipate opposition and formulate plans that mitigate risk. RDS facilitates the identification of HVIs by uncovering explicit links between individuals, organizations, and events using Relationship Discovery Format (RDF) triples (noun, predicate, and object) as input mapped to a supporting ontology. Output generated by RDS consists of graphed relationships in the form of networked nodes and links with corresponding network measures of centrality (Moss, Thomas, and Mittrick, 2011).

Measures of centrality are used to assess entities' social strength and importance within a network. Primary measures of centrality include betweenness, closeness, and degree (Freeman, 2004). An entity with high betweenness acts as an information broker or gatekeeper in a network by connecting entities that would otherwise not be connected. This connectedness greatly influences the flow of information between individual and groups of entities. High closeness characterizes an entity within the network that through their pattern of direct and indirect relationships is able to access a large number of other entities within the network faster than any other entity. The entity with the highest degree is the network "hub"; the most connected and active entity of the network.

Centrality measures used in conjunction with RDS's visual representation of the network assist intelligence analysts in fusing information necessary to detect gatekeepers and hubs of a social network. This emphasis on the fusion of visual analytics and quantitative measures situates RDS as a promising automated analysis tool for novice intelligence analysts typically found in Company Intelligence Support Teams (COISTs), though it would also be useful to experienced analysts.

Intelligence analysts at the company level usually do not possess the same level of expertise and training as higher echelon counterparts and require assistance from automated tools. In the current operating environment, COISTs operate on the ground gathering and determining the significance of intelligence, often without the assistance, analysis and filtering of higher-level intelligence staff support. This small-team focused intelligence enables the company to maintain situational awareness while conducting daily activities such as patrols, engagements, and combat logistics (Flynn, Pottinger, and Batchelor, 2010).

Development of the RDS platform as an automated social network analysis tool for COISTs is nearing its fifth year. This report documents an investigation and evaluation of the current state of development. The next section of the report, section 2, describes a data set used to evaluate the RDS platform. The data set is an unclassified collection of fictitious text messages revolving around a hidden terrorist plot. The data set called, Ali Baba, was originally created by the National Security Agency (NSA) and then modified and manually verified by CISD engineers to meet the requirements of the RDS evaluation.

Section 3 of the report details the procedure of a user test of RDS. Initially, a research protocol for a human subject usability experiment was designed and approved. Unfortunately, system issues with the RDS platform were discovered, making the designed human subject experiment too problematic to conduct. As an alternative, an informal user test was performed to evaluate the current state of RDS. The participants of the user test were CISD staff generally unfamiliar with the social network analysis capabilities of RDS and the Ali Baba data set.

Section 4 discusses the results of the user test. CISD staff participating in the user test played the role of “novice intelligence analysts” similar to analysts working in COIST. The goal for the participants was to extract important information about a terrorist plot from the Ali Baba data set. Time taken to identify plot details and accuracy of the identified information were recorded as an assessment of each participant’s level of understanding of the data. Evaluation of RDS is in the form of observations generated during the user test about system responsiveness and reliability, and suggestions for improved functionality.

The summary of the user test results and observations in section 5 illuminates ongoing development challenges with the RDS platform as a COIST-level analysis technology. These challenges are discussed within the context of RDS system revisions and learning outcomes. Section 6 consolidates the scientific and technical contributions of RDS to the field of social network analysis.

2. Ali Baba Data Set

An unclassified fictitious collection of text communications containing a hidden terrorist plot was needed to evaluate the social network analysis capabilities of RDS. A data set created in 2003 for NSA by Mark Jaworoski and Steve Pavlak, called Ali Baba, was selected as a starting point. This data set contains 752 text messages recording the actions of a suspected terrorist network located in England intent on bombing a water treatment plant.

The Ali Baba data set consists of two collections of messages: a structured version using an Excel spreadsheet format and an unstructured version formatted as scrap files. A manual cleaning process was performed to correct message content misalignment across the two versions. Mittrick, Roy, Kase, and Bowman (2012) discuss the cleaning and refinement process of the Ali Baba data set; the generation of RDF triples and an ontology as input for the RDS system; and the validation of truth products using concept mapping techniques. Readers are requested to refer to Mittrick et al. (2012) for complete background information on the Ali Baba data set used in the testing and evaluation of the RDS system.

3. Informal User Test

To test the current development state of RDS, an informal user test was conducted using CISD staff relatively unfamiliar with both the social network analysis capabilities of RDS and the Ali Baba data set. The CISD participants, in the role of "novice intelligence analysts," qualitatively rated RDS in performing social network analysis functions on the Ali Baba data set. Comments and observations were collected across participants and observers involved in the study. A set of questions was prepared to evaluate the level of understanding the participants gained about the ground truth of the Ali Baba data set.

Because the user test was informal in nature, it was composed of three simple activities. Activity 1 consisted of the participant (herein called analyst) observing the system in use by a knowledgeable user (herein called experimenter). Activity 2 consisted of the analyst operating the system with guidance from the experimenter. In Activity 3, the analyst used the system, unassisted, to analyze the Ali Baba data set. The goal for the analyst was to uncover the ground truth contained in the Ali Baba data set within a 2-h time period and answer a set of questions pertaining to details about the ground truth. Results of the user test are discussed in terms of observations suggesting reasons for the somewhat low performance ratings of RDS, and highlighting those aspects of the system the analysts and observers want to see improved.

During Activity 1 the experimenter demonstrated a method of analysis to the analyst. The demonstration involved the use of four windows included in the RDS interface: Investigator window, Database Controller window, Centrality Metrics window, and Ontology Navigator window. The default view of the RDS interface is shown in figure 1.

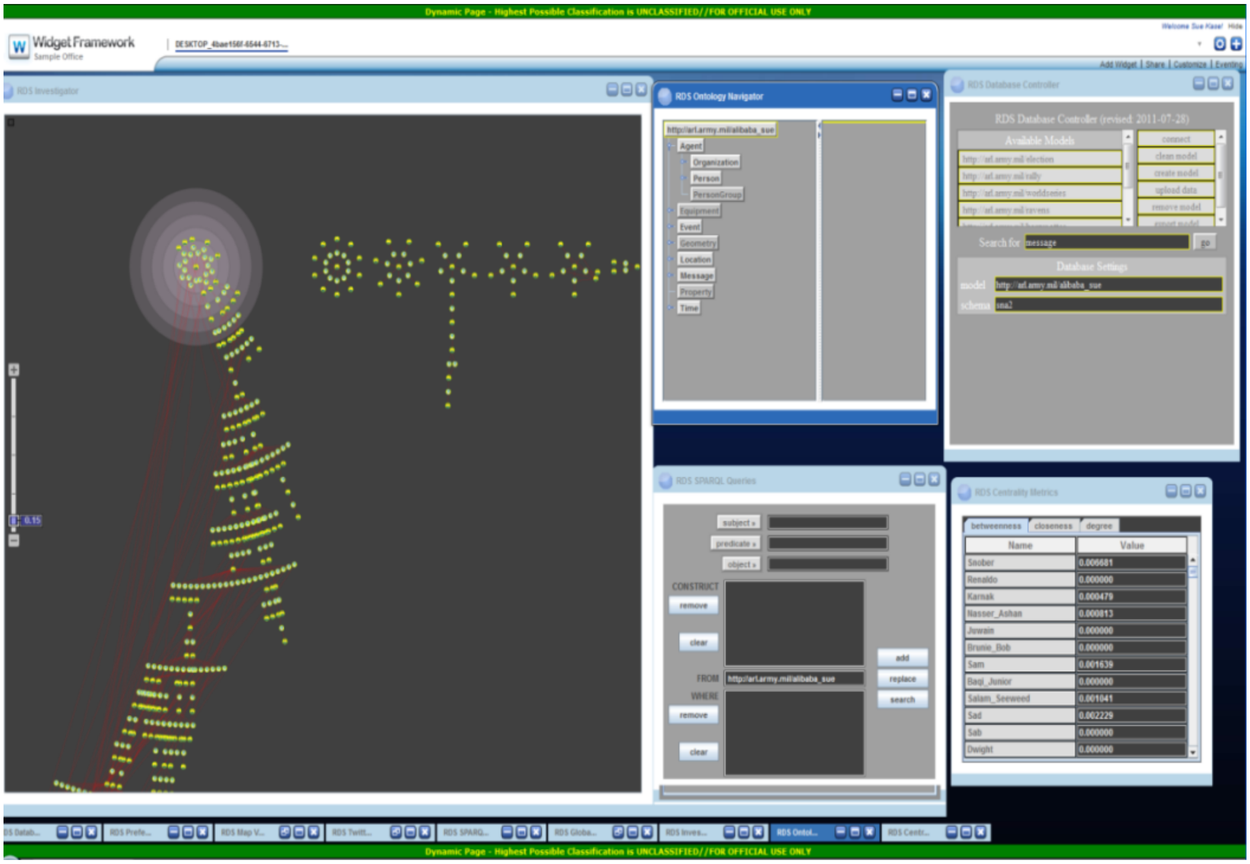

Figure 1. RDS interface with main windows open in the default view.

Basic information about the test data set was explained to the analyst, such as the types of communications contained in the data (i.e., intelligence information reports, police reports, and tactical reports). RDS extracts information from the contents of communications and categorizes it according to the ontology currently loaded into the system. The experimenter showed the analyst the Ontology Navigator window and how to expand the ontology levels. In order for RDS to parse the contents of the communications, "messages" must be loaded into the Investigator window. When the experimenter typed "message" in the "Search for" textbox in the Database Controller window, all the communications were displayed as message icons in the Investigator window (figures 2 and 3).

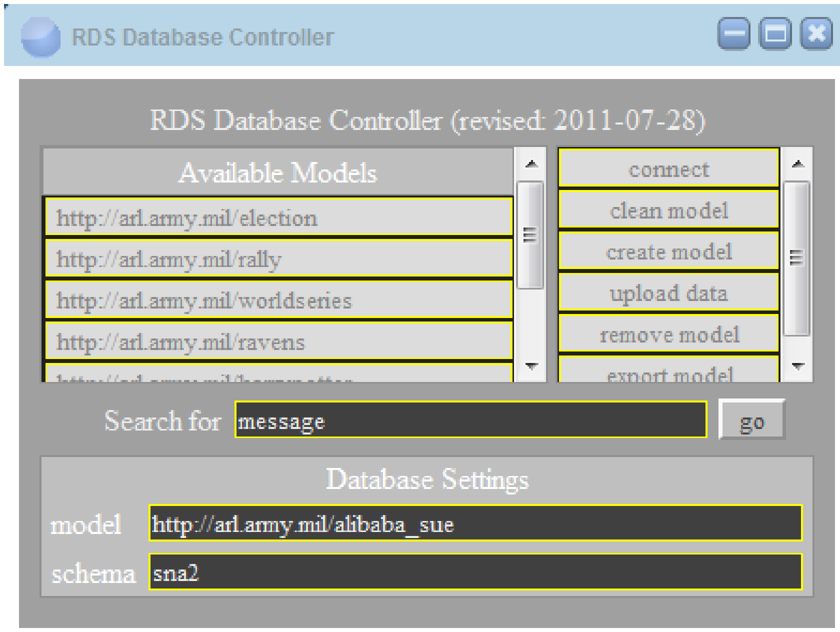

Figure 2. The “Search for” textbox with “message” typed into the textbox field.

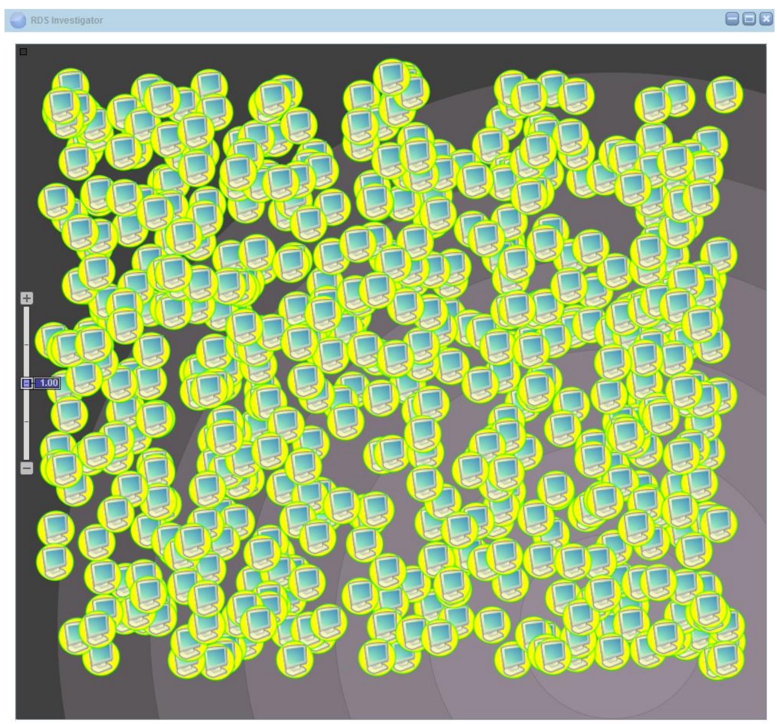

Figure 3. The Investigator window displaying message icons.

It takes RDS time to parse the messages as displayed by a “processing” bar in the bottom left corner of the Investigator window. The analyst was asked to wait while RDS processed information (when the processing bar was active). After the messages appeared in the Investigator window, the experimenter showed the analyst how to open one of the messages by holding down the shift key and hovering the mouse over a specific message icon. Each opened message appeared in a movable window divided into fields corresponding to the ontology (figure 4).

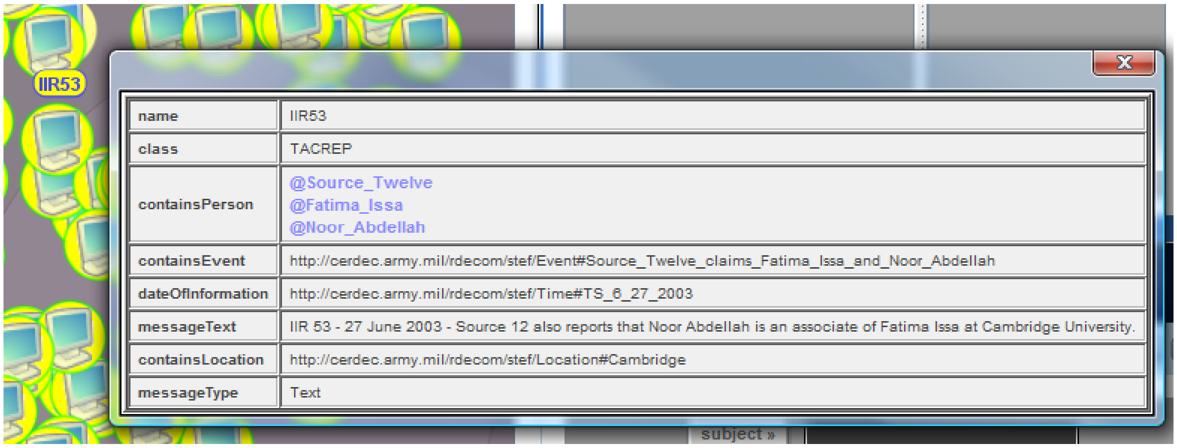

Figure 4. An open message with several ontology fields populated.

The experimenter noted the correspondence of the ontology to the parsed message fields in the message window. If a message window was closed (by clicking on the red x in the upper right corner), it could not be reopened again without restarting RDS. The experimenter showed the analyst how to organize open message windows on a second display monitor.

Next, the experimenter demonstrated the linking and selection menu accessed with a right-click in the Investigator window (figure 5). Available link options are colored dark gray. The experimenter used the menu to link the messages by person (right-click > person > containsPerson). Each reference to a person in the messages is represented by a person icon. Links are represented as red lines connecting the persons referenced in a message to the actual message icon. The experimenter hovered the cursor over one of the links showing the “containsPerson” link label (figure 6) and opened the message to show the persons parsed from the contents.

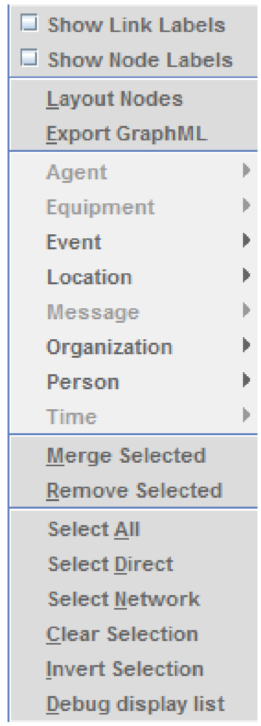

Figure 5. The link and selection menu.

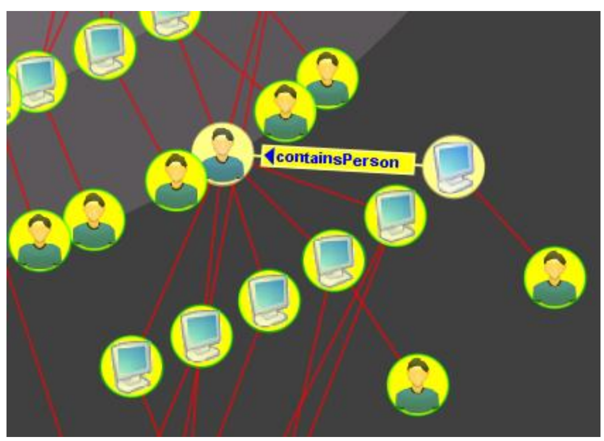

Figure 6. Messages linked (red lines) by “containsPerson.”

Often RDS does not calculate the routes for the links and this is evident by red link lines all running parallel off the right edge of the Investigator window (figure 7). This is a known error in the system. When this error occurred, the analyst was directed to do the linking process again using the right-click menu. When the links are correctly calculated, a “calculating routes” processing bar appears (figure 8).

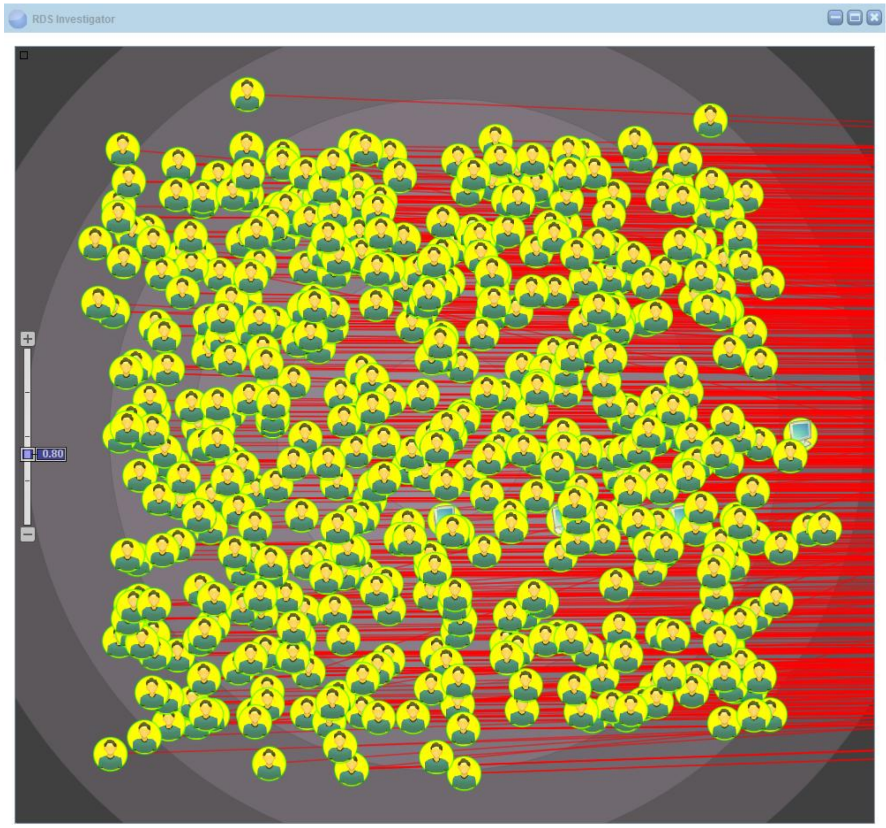

Figure 7. System error when link routes are not calculated correctly.

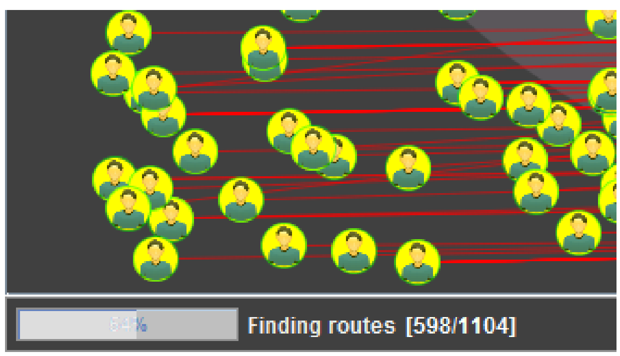

Figure 8. Processing bar showing link routes being calculated.

At this point in Activity 1, the experimenter demonstrated how to navigate around the Investigator window: hold down the left mouse button and move the cursor to pan; and click along the vertical bar on the left edge of the window to zoom. The experimenter pointed out how some person icons have many links connecting them to messages and other persons. The number of links is a measure called "degree." Degree is a network centrality measure for the number of links or network ties a person has.

When RDS completes a linking operation, the Centrality Metrics window is populated with values (figure 9). RDS calculates three centrality metrics: betweenness, closeness, and degree. A simple definition of each metric was explained to the analyst. The experimenter demonstrated how to access each metric tab and order (ascending, descending) the currently active metric values.

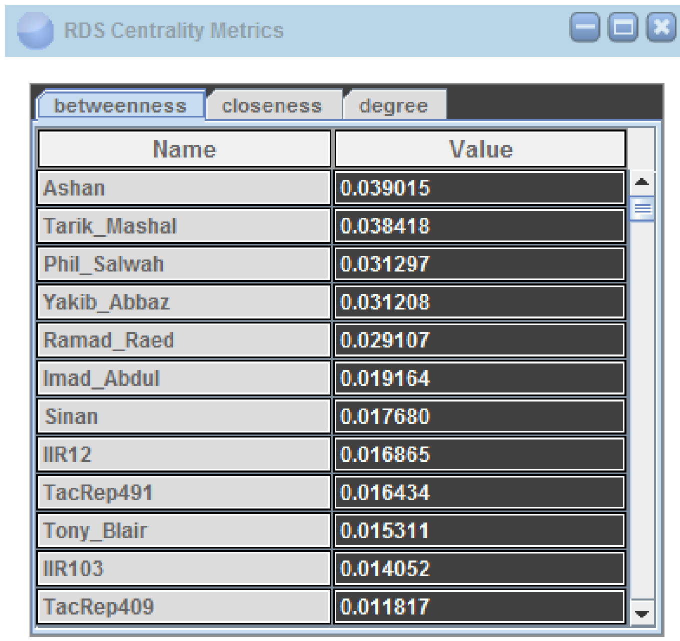

Figure 9. The Centrality Metrics window with the "betweenness" tab activated.

By default, objects (in this case, messages and persons) loaded into the Investigator window are in "selected" mode. The experimenter demonstrated how to deselect objects using the right-click menu. Icons are yellow in color when selected, and gray when deselected.

The "Search for" textbox used previously to load messages is also a search feature. The experimenter showed how to search for the person with the highest degree value as listed in the Centrality Metrics window. When found, the person's icon becomes selected (changes color to yellow). This search feature was demonstrated a second time using a person with a high betweenness value. Some of the messages linked to this particular person were opened, read, and moved to the second display monitor for later reference. The experimenter explained that the centrality metrics values can be used as a starting point for reviewing and organizing message content in the process of analysis.

The second part of the user test, Activity 2, consisted of the analyst using RDS to repeat the above procedure (messages linked by persons) from start to finish including using the search feature to locate persons with high centrality metric values. This activity gave the analyst an opportunity to use the system with the experimenter present to offer assistance.

In the last part of the user test, Activity 3, the analyst was instructed to analyze the Ali Baba data set and identify the ground truth. RDS does not have a scratch pad feature for note taking during the analysis process. As a substitute, a Notepad file was opened and titled “Data analysis scratch pad.” The experimenter explained that this file can be used to jot down trends and ideas pertaining to the ground truth as they develop during the analysis process. The Notepad file also contained questions for the analyst to answer by the end of their two-hour analysis session. The Notepad file contained the header “Determine the terrorist plot to be carried out” and table 1 shows the list of seven questions.

Table 1. List of questions for the analyst to answer at the end of the analysis session.

1. Who are the key players?
2. What are the roles of the key players?
3. What is the plot?
4. How will the plot be carried out?
5. What is the target?
6. Where is the target?
7. What is the motivation behind the plot?

This set of questions was used to evaluate the level of the analyst’s understanding of the ground truth contained in the Ali Baba data set. Both time to identify the ground truth and accuracy of the ground-truth identification were considered when assessing the analyst’s level of understanding.

4. Results

Several CISD staff acting in the role of novice intelligence analysts participated in the user test. This section first describes a particular analyst’s note taking process. Next, the analyst’s answers to the set of questions on the ground truth are scored. Observations collected during the user test are discussed. Lastly, suggestions for improvement in the areas of system reliability and responsiveness, and functionality are noted.

4.1 Analyst Results

Notes and results from an analyst, one of several analysts participating in the user test, are presented in table 2. This analyst’s use of RDS was of interest because an alternative analysis method was attempted during Activity 3 of the user test. This alternative analysis method, not demonstrated by the experimenter during Activity 1, offered both advantages and disadvantages in uncovering the ground truth as discussed later in this section.

Table 2 shows some of the notes recorded by the analyst in the Notepad file. The notes show probable persons of interest identified during the beginning stages of Activity 3. At this point, the analyst identified five (Ashan, Tarik Marshal, Phil Salwah, Yakib Abbaz, Imad Abdul) of the ten most important persons involved in the terrorist plot. Under each person's name the analyst recorded notes from message content along with the corresponding message identifiers.

Table 2. Notes recorded by an analyst during Activity 3 of the user study.

Ashan
- Friend Abdul-Mudi wanted to take flying lessons (TacRep411)

Tarik_Mashal
- Asked Ali Hakem if he was available next year. Ali Haken suspected weapon maker (TacRep03)

Phil_Salwah
- Sent to England to continue Jihad against British Govt. Taliban (IIR80). Knows Detainee Madison and Massad

Yakib
- Told Sifiy he would use his skills for Jihad (TacRep 230)
- Connected to Source_Seven, who said plot to kill hundreds around London (IIR33)
- Connected to Salam Seeweed - financier (TacRep25)
- Connected to Datainee Kimberly through Salam Seeweed and/or Wadi Wakiweed (IIR78)
- Connected to Imad_Abdul

Abdul
- Ruthless tactician and organizer (IIR56)
- Wealthy
- Reported to Ali Baba that all preparations for baking the cake were proceeding. Imad_Abdul works for Ali Baba (TacRep50)

Before the user test session ended, the analyst answered the seven questions about the terrorist plot. Table 3 shows the analyst's answers to these questions.

Table 3. Analyst's answers to the seven questions about the terrorist plot.

1. Who are the key players?: Yakib_Abbaz, Imad_Abdul, Tarik_Mashal, Ramad, Ali_Baba

2. What are the roles of the key players?: Imad_Abdul is the organizer, Ramad is to set-up a diversion, Ali Baba is the leader

3. What is the plot?: Making a bomb.

4. How will the plot be carried out?: Ali Hakem bomb maker

5. What is the target?: Water Treatment Facility

6. Where is the target?:

7. What is the motivation behind the plot?: Golan Heights

For question 1 on key players, the analyst listed a slightly different subset of persons (Yakib Abbaz, Imad Abdul, Tarik Marshal, Ramad, Ali Baba) than appeared in the above notes. Here Ashan was not included but Ramad Read was. Both Ashan and Ramad were distracters. Phil Salwah, a recruiter, was not included, but Ali Baba, the leader of the group, was included. The solution for question 1 is 10 key players. The analyst listed 5 of the 10. The analyst's answer for question 2 on the roles of the key players was incomplete with only 3 of the 10 specified key persons being assigned roles. Nevertheless, the roles of the three persons listed were correct (Imad Abdul is the organizer, Ramad is to setup a diversion, Ali Baba is the leader). For question 3 the analyst identified the plot as "making a bomb" with Ali Hakem as the bomb maker (question 4). Ali Hakem was actually a computer hacker initially asked to participate in the plot but in the end was not involved. Hosni Abdel was the bomb maker. For question 5 about the target, the analyst answered correctly with "water treatment facility," but could not identify the location (question 6). For the last question on the motivation of the plot, the analyst's answer of "Golan Heights" was unexpected, as well as incorrect, as it appeared only once in a message and was not searchable or listed as a location instance in the ontology. This, in fact, was due to an error in creating the data set triples.

To calculate a score we counted questions 3 through 7 worth 1 point because they required only a single answer. Questions 1 and 2 essentially have 10 subparts because there are 10 key players involved in the ground truth of the data set. We counted each of these questions as worth 5 points with each subpart worth ½ a point each. In total, the questions are worth 15 points. An analyst's score is calculated as their number of correct points divided by the total number of points.

The analyst's score calculated for the answers shown in table 3 is 6 out 15 points: two of the 1-point questions were correctly answered; question 1 with 5 out of the 10 subparts correct scored 2.5 points; and question 2 with 3 out of the 10 subparts correct scored 1.5 points.

The analyst spent approximately 2 h completing Activity 3 of the user study. The analyst's score (6/15) is slightly below average. Considering the amount of RDS training and time allocated for extracting the ground truth, the low score was not surprising. It was hoped that the RDS-assisted analysis would result in at least above-average scores. This could be the result of the minimal amount of RDS training presented to the analyst, or evidence that RDS is not assisting the analyst as expected. Training time and type of training are variables to be adjusted if additional rounds of the user study are planned.

4.2 Observations

Observations from the experimenter and a second individual acting as a participant observer are discussed below. These observations are in reference to the analyst's user test during the three activities and results as discussed previously.

During Activity 1, the analyst appeared to be attentive, but asked few questions during the demonstration. The most significant observations by the analyst were centered on the processing lags inherent in RDS, for example, when parsing message content and calculating link routes. After the analyst's initial comment that RDS is "slow," from then on, the analyst patiently waited until the processing bar disappeared before proceeding on with the next operation. In addition to processing lags, RDS would sometimes crash or lock-up. When this occurred and the analyst navigated to the Investigator window, nothing would happen if a search or link operation was requested. In this case, the system window had to be closed and a restart of RDS was necessary.

During Activity 2, when the analyst was asked to repeat the demonstration analysis, some assistance was offered by the experimenter at points when the analyst appeared to forget how to execute a specific operation, for example, how to open a message or deselect nodes. For the most part, the few operations needed to accomplish a basic analysis in RDS appear to be quickly learnable.

The analyst primarily used the search textbox for displaying collections of ontology objects as well as searching for specific nodes; the Investigator window for visually inspecting the network structure; and the Centrality Metric window for obtaining information on persons with high centrality values. The other many windows offered by RDS were not used by the analyst during the user test probably because their purpose was not demonstrated by the experimenter during Activity 1 and their functionality was not needed to analyze the test data set.

During Activity 3, the analyst switched to a different analysis method primarily using the Ontology Navigator window. This was of interest with details as follows. The analyst displayed messages in the Investigator window and then linked the messages by "containsPerson." This populated the Centrality Metrics window. This is the same analysis process as demonstrated by the experimenter during Activity 1. The difference is, at this point, the analyst recorded some of the names of the persons with high betweenness and degree values in the Notepad file. Then the analyst selected everything in the Investigator window and removed it, leaving an empty

Investigator window. The purpose of displaying the messages and linking them by person was only to obtain the centrality metrics values for later reference.

With several high-valued persons from the Centrality Metrics window jotted down in the Notepad file, the analyst then used the Ontology Navigator window to find those persons listed under Agent > Person > Terrorist of the ontology (figure 10). Although person instances are alphabetically ordered in the Ontology Navigator window, finding a specific individual is time consuming due to the total volume of persons contained in the data set. Once located in the ontology, clicking the person's name adds that person as a node in the Investigator window. When the person is selected the link menu can be used to display the messages referencing that person.

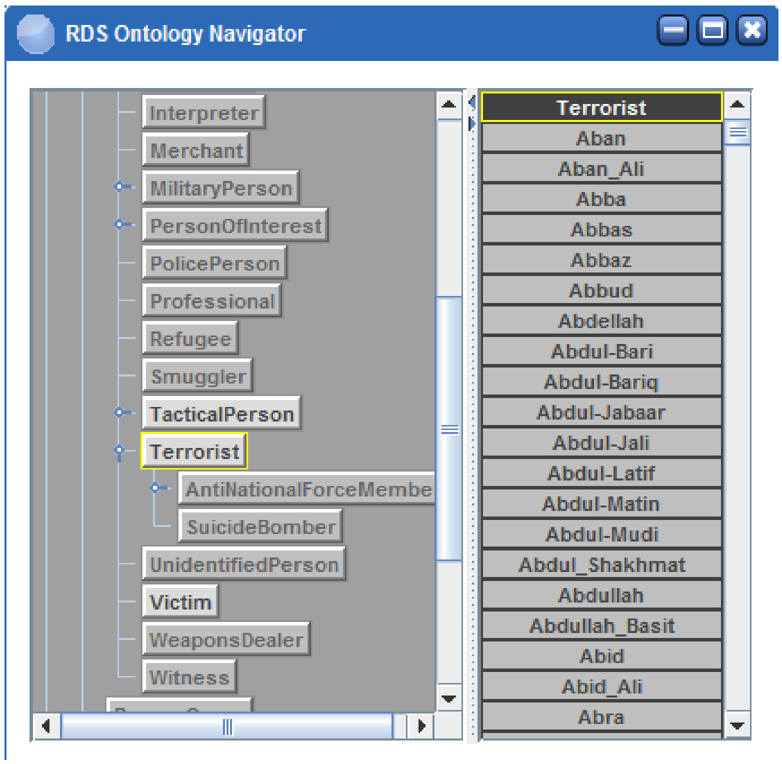

Figure 10. The Ontology Navigator window showing person instances.

Using this type of analysis method, an analyst can build a network containing only the persons of interest and their associated messages (or events, locations, and other ontology objects). Analyzing only a small subset of the entire network is obviously easier than analyzing the entire network, which can be effortful and sometimes intimating. In addition, the system processing lags can be reduced considerably depending on the size of the subset of data visualized in the Investigator window.

The disadvantage of this type of analysis method is the subset of the network chosen by the user to visualize and analyze must include some bit of information leading to the discovery of the ground truth; otherwise, the ground truth may remain hidden because of its exclusion from the

Investigator window and thus the analysis. For example, the analyst's surprising answer to the plot motivation question (question 7) of "Golan Heights" probably resulted from the analyst selecting only a very small subset of messages with one of the messages referencing Golan Heights. Having read through the message content of only the displayed subset, the analyst erroneously concluded Golan Heights was the motivation for the terrorist plot.

4.3 Need for Improvement

Overall, analysts' comments collected during the user test suggested RDS needs improvement in the areas of system functionality, documentation, and system reliability and responsiveness. This could be the result of the limited system demonstration and training the analysts received before the user test, but may also reflect the actual need to improve the system. On a positive note, RDS was evaluated as more effective than manually analyzing the test data set (e.g., drawing a concept map).

There were several suggestions related to usability. As mentioned previously, RDS does not include a scratch pad feature. Analysts have to remember what they have found or use external support such as a text editor (e.g., Notepad, Word) or paper and pencil to reduce short-term memory load. The use of a supporting application for note taking requires a second display screen. For example, the analyst typed names of persons with high centrality metrics in the NotePad file (open on a secondary display screen) for reference when searching through the ontology instances (with RDS open on the primary display screen).

If a copy and paste of a subset of the contents of the Centrality Metrics window was possible, the analyst could paste the names of the top persons from the metrics window into the NotePad file instead of retyping the names. In addition, a copy and paste of segments of important message content could serve as useful notes. Currently, a copy and paste of message content in a message window results in the message window permanently closing. Message windows always stay in the foreground covering the other RDS windows. They are movable but not resizable, and if closed, cannot be reopened again. If an analyst wants to leave some messages open for later reference, once again, a second display screen is required for displaying and organizing the messages.

The textbox in the Database Controller window serves dual purposes: to load and display ontology objects in the Investigator window and search for specific nodes already displayed in the Investigator window. This search feature was frequently used by the analysts during the analysis process. Currently, the search feature operates under a 100% match criteria and does not support partial matching. Partial matching would be helpful during the analysis process especially because the system does not offer a scratch pad feature for recording names discovered during the analysis. On several occasions, the analyst attempted to search on only part of a person's name, first or last—with the search resulting in no match.

If a search is executed and the searched for node is found, that node changes color to indicate a “selected” state. If a second search is performed and the node found, the previously selected node changes back to an unselected state while the new search result becomes selected. It is not possible for successive search results to stay in a selected state. For example, if an analyst wants to search for the top three persons with high degree values and then read only the messages associated with those three persons. If those three persons would stay selected, they could act as reference points during the analysis.

On the other hand, if a search is executed and the searched for node is not found, the analyst does not know that the search failed unless they visually scan the entire contents of the Investigator window (sometimes involving many pans and zooms) looking for a selected node. The addition of a “found” or “not found” search result indicator would be beneficial to the analyst.

Currently, RDS lacks documentation including a user’s guide or system/developers manual. Experienced users of RDS either learned how to use the system by trial and error or obtained assistance from other RDS users. A user’s guide explaining the basic functionality of the system including simple analysis examples is required if RDS development continues.

The greatest opportunities for improvement lay within system reliability and responsiveness. The lags in system processing during functions such as adding properties for instances, queuing changes to graph links, and calculating link routes, practically render the system unusable outside of development and testing. RDS users must wait until the processing bar in the lower left corner of the Investigator window disappears before proceeding with their next intended operation. This wait time is disruptive to analysts’ thought processes and slows the overall analysis substantially. Unfortunately, the processing lags increase with the size of the data set being analyzed.

Occasionally, the system neglects to calculate the link routes altogether. An inexperienced RDS user probably would not detect the error assuming link routes have been calculated. There is no visible error message only a mass of red lines running off the right edge of the Investigator window. When link routes are calculated, the Investigator window shows an orderly arrangement of nodes in concentric circles with red link lines connecting the nodes. A new RDS user must be directed to specifically look for the processing bar with the label “calculating routes.” Sometimes the link selection operation must be repeated several times using the right-click menu until the system proceeds with calculating the link routes.

5. Discussion

The intent of RDS is to synthesize collected human intelligence (HUMINT) information expressed in a variety of text message formats into RDF triples that serve as data input for the analysis components of the system. The system, in turn, enables the data mining of the RDF triples via a supporting ontology for interesting relationships among entities such as people, organizations, and events. The data mining techniques used within RDS include social network analysis, graph theory, and logistic regression. Additional system information on the RDS platform can be found in Moss et al. (2010).

The space in which social relationships are discoverable in RDS is static for a given analysis and defined by the ontology that lists all the potential classes and subclasses of entities. The ontology can be extended to account for a greater variety of relationships. RDS development emphasizes the visual display of social relationships in the form of nodes and links consistent with standard measures of network centrality such as betweenness, closeness, and degree. In theory, this automated display of relationships and quantitative network measures would guide an analyst more quickly to important nodes or links, with details retrievable by drilling down to see the actual message content.

For testing and evaluating RDS, a suitable set of data was needed. The data set had to be sufficiently rich to support the analyst's task of extracting basic knowledge of a terrorist cell's composition and plans from a large set of text messages. Within the set of messages, there must be a scenario storyline or ground truth for the analyst to discover, and in an evaluation situation, one that offers reasonable measures of analyst and system performance. The Ali Baba data set originally created by NSA was selected as the test data set. This data set was extensively cleaned and modified with the ground truth verified by a concept map method of analysis.

When considering the current development state of RDS, the potential for formal experimentation or even a quantitative comparison of competing software systems offering similar functionality appeared somewhat limited. Although the development of RDS spanned nearly five years, the current state of the system can be categorized as a "working prototype" and is not portable to a laptop computing environment. This portability issue poses a serious challenge for conducting formal human subject experimentation. Potential subjects would need to be brought into the division's computer laboratory where RDS was installed and running on a desktop computer.

As an alternative to formal experimentation, a simple user test was designed. The user test qualitatively evaluated the performance of RDS by recording, for example, time to uncover a specific chunk of information or the proportion of ground truth obtained from the data set. The focus of the test is some component of understanding afforded to the analyst by RDS and this, in turn, would assist in assessing the value of the system.

In the previous sections, we reported details of the user test procedure along with results from one of the analysts participating in the study. The understanding the analyst gained about the Ali Baba data set was evaluated using a set of questions pertaining to details of the ground truth. A score was calculated based on the analyst's correct answers divided by the total number of points across all questions. The example analyst's score was slightly below average. In addition to question-answer evaluation, observations and feedback from the experimenter, participant observers, and the analysts themselves were recorded. This feedback offered more in depth concerns related to system functionality, documentation, and system reliability and responsiveness.

Of greatest concern are the lags in system processing during some linking functions. System responsiveness appeared to deteriorate as the number of messages displayed in the Investigator window increased. The Ali Baba data set is considered rather small compared to the size of a real-world set of communications an intelligence analyst would encounter. When analyzing the Ali Baba data set, RDS showed poor responsiveness when a link operation required route calculations, sometimes not performing the route calculations at all or requiring a system restart. Some of the secondary concerns included the need for a second display screen for organizing open message content because message windows cannot be closed and reopened or put in the background; lack of a scratch pad feature for recording evolving analysis ideas (also requiring a second display screen for use of a supporting text editor); limited copy and paste functionality; lack of user documentation; no “found” or “not found” search indicator; and partial matching not implemented for search.

At the time of conception, RDS innovatively filled a technology gap, offering novice intelligence analysts a system-assisted situational understanding of people, organizations, and events.

Unfortunately, lengthy development time combined with lack of design specifications allowed other systems with similar functionality to be developed and fielded successfully. In addition, the push to cloud computing infrastructures, for example, the integration of analytical applications to the DCGS-A Standard Cloud (DSC), emphasizes the need for portability. Portability is especially important as the intended user group for RDS is COIST analysts in the field. Portability was not considered during RDS development up until now.

Although the user test of RDS was informal in nature, it brought to light many usability-related issues and system-related challenges. The effect these challenges on COIST analysts would be considerable. It might be that the current development trajectory of RDS is a better fit to upper-level intelligence analysts such as those employed at the Battalion S2 level. RDS offers many additional analysis features not used in the user test presented in this report. This test was designed to evaluate RDS as tool for novice intelligence analysts with little to no analysis experience. At this point in time, we have decided to temporary suspended further development of the RDS platform until a future direction and systematic plans for revision can be determined.

The disappointing outcome of RDS as a potential COIST technology does provide several learning-related opportunities. The current version of RDS, as an in-house analysis application, can be used as a “learning prop” for studying common computing issues such as scalability, portability, architecture integration, new data and model frameworks, and development for cloud computing environments. Additionally, portions of RDS can be reused and modified for development of in-house analytic and data conversion tools such as the system’s components for processing text messages as RDF triples and the algorithms for calculating network centrality measures. The most noteworthy outcome of RDS is the refined Ali Baba data set. The effort spent modifying and verifying the data set has already proved useful in many applications. Recently, the Ali Baba data set was shared with several Government partners and small businesses as a test and evaluation measure for social network analysis applications under development.

6. Conclusion

The RDS platform is an integrated set of Web-enabled, service-oriented applications designed to assist in the analysis of soft intelligence data. An investigation was conducted to assess the current development state of RDS as a potential COIST technology. Typically, intelligence analysts at the company level are novices with little to no analyst training or experience. RDS offers the capability to automatically construct social networks from text-based messages, visually display the social network relationships and calculate the associated network measures of centrality. With these capabilities, RDS could assist COIST analysts in organizing and extracting key information about terrorist organizations and their activities.

To evaluate RDS as a COIST technology, an informal user test was performed using CISD staff playing the role of novice intelligence analysts. The goal for these analysts was to uncover the ground truth contained within a test data set of text messages. A data set originally developed by NSA, called Ali Baba, was cleaned and modified for testing purposes. The Ali Baba data set consists of over 700 text messages recording the actions of a fictitious terrorist network planning an attack on a water treatment plant.

Usability issues and system reliability and responsiveness difficulties uncovered during the user test indicate RDS is not suitable as a COIST-level social network analysis tool. The analysis capabilities offered by RDS may be more appropriate for use by experienced intelligence analysts at the Battalion S2 level. At the present, continuation of RDS development has been temporary suspended until plans for revision are formalized.

7. References

Flynn, M. T.; Pottinger, M.; Batchelor, P. D. *Fixing Intel: A Blueprint for Making Intelligence Relevant in Afghanistan*. Center for a New American Security Working Paper, Jan 4, 2010.

Freeman, L. C. *The Development of Social Network Analysis: A Study in the Sociology of Science*. Vancouver: Empirical Press, 2004.

Mittrick, M.; Roy, H.; Kase, S.; Bowman, E. *Refinement of the Ali Baba Data Set*; ARL-TN-0476; U.S. Army Research Laboratory: Aberdeen Proving Ground MD, 2012.

Moss, G.; Thomas, M.; Mittrick, M. *Fiscal Year 2010 Status Report for Advanced, All-Source, Fusion Technology Program Annex*; ARL-TR-5489; U.S. Army Research Laboratory: Aberdeen Proving Ground MD, 2010.

List of Symbols, Abbreviations, and Acronyms

AO	area of operation
ARL	U.S. Army Research Laboratory
A2SF	Advanced All Source Fusion
CERDEC	Communications-Electronics Research, Development, and Engineering Center
CISD	Computational Information Sciences Directorate
COIST	Company Intelligence Support Team
DSC	DCGS-A Standard Cloud
HUMINT	human intelligence
HVI	high-valued individuals
I2WD	Intelligence and Information Warfare Directorate
NSA	National Security Agency
RDF	Relationship Discovery Format
RDS	Relationship Discovery Service
SNA	Social Network Analysis
TIFB	Tactical Information Fusion Branch
TPA	Technical Program Annex

NO. OF COPIES	ORGANIZATION
1 ELEC	ADMNSTR DEFNS TECHL INFO CTR ATTN DTIC OCP 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	ARMY G2 ATTN D WALSH 4812 EXECUTIVE DR GAINSVILLE VA 20155
1	US ARMY CERDEC I2WD ATTN RDER IWP D PORTER 6003 COMBAT DR ABERDEEN PROVING GROUND MD 21005
1	MULTISOURCE INFORMATION FUSION ATTN RESEARCH PROFESSOR (EMERITUS) J LLINAS 315 BELL HALL NORTH CAMPUS STATE UNIVERSITY OF NEW YORK AT BUFFALO BUFFALO NY 14260
1	U.S. ARMY RSRCH LAB ATTN RDRL CII C E K BOWMAN BLDG 390A RM 112 ABERDEEN PROVING GROUND MD 21005
10	US ARMY RSRCH LAB ATTN RDRL CII C M MITTRICK ATTN RDRL CII C S KASE ATTN RDRL CII C A NEIDERER ATTN RDRL CII C B BODT ATTN RDRL CII C C HANSEN ATTN RDRL CII C E HEILMAN ATTN RDRL CII C F S BRUNDICK ATTN RDRL CII C H ROY ATTN RDRL CII C J DUMER ATTN RDRL CII C J RICHARDSON ABERDEEN PROVING GROUND MD 21005
1	US ARMY RSRCH LAB ATTN RDRL CII C M THOMAS BLDG 321 RM 1B ABERDEEN PROVING GROUND MD 21005
1	US ARMY RSRCH LAB ATTN RDRL CII C T HANRATTY BLDG 321 RM 4A ABERDEEN PROVING GROUND MD 21005
8	US ARMY RSRCH LAB ATTN IMNE ALC HRR MAIL & RECORDS MGMT ATTN RDRL CII A S H YOUNG ATTN RDRL CII B BROOME ATTN RDRL CII B L TOKARCIK ATTN RDRL CII B R WINKLER ATTN RDRL CII T V M HOLLAND ATTN RDRL CIO LL TECHL LIB ATTN RDRL CIO LT TECHL PUB ADELPHI MD 20783-1197